

1     Claims

2  
3             What is claimed is:

4  
5             1. A method of managing a file system for a file server, comprising the steps of:  
6             receiving a file operation that signals a reservation operation for a file having a  
7     file size;  
8             computing a number of blocks needed to be reserved to accommodate the file; and  
9             reserving a number unallocated blocks in the file system equal to the number of  
10    blocks needed to be reserved to accommodate the file.

11  
12            2. A method as in claim 1, wherein the file system uses a write anywhere file  
13    system layout.

14  
15            3. A method as in claim 1, wherein the file operation that signals the reservation  
16    operation is a zero length write request.

17  
18            4. A method as in claim 1, wherein the file operation that signals the reservation  
19    operation includes a parameter that specifies the file size.

20  
21            5. A method as in claim 1, wherein the step of computing the number of blocks  
22    needed to be reserved to accommodate the file further comprises:

23            determining a total number of direct and indirect blocks needed to accommodate  
24    the file size; and

25            subtracting a total number of blocks already allocated for the file and a total  
26    number of cached unallocated blocks for the file from the total number of direct and indirect

1 blocks needed to accommodate the file size.

2  
3 6. A method as in claim 1, wherein the step of reserving the number of  
4 unallocated blocks in the file system equal to the number of blocks needed further comprises:  
5 setting a flag in an inode for the file that indicates blocks have been reserved for  
6 the file; and  
7 incrementing a reserved block count in a file system information block by the  
8 number of blocks needed, the reserved block count indicating how many unallocated blocks have  
9 been reserved for files in the file system.

10  
11 7. A according to claim 1, further comprising the step of checking that a number  
12 of available blocks in the file system is greater than the number of blocks needed to be reserved  
13 to accommodate the file, wherein an error is returned in a case that the number of available  
14 blocks is less than the number of blocks needed.

15  
16 8. A method as in claim 7, wherein the number of available blocks in the file  
17 system is determined by subtracting a number of allocated blocks, a number of cached  
18 unallocated blocks, and a number of reserved blocks from a total number of blocks in the file  
19 system, and adding a number of reserved cached unallocated blocks.

20  
21 9. A method according to claim 1, further comprising the step of checking that  
22 the number of blocks needed to be reserved to accommodate the file does not exceed a remainder  
23 of a quota for an owner of the file, wherein an error is returned in a case that the number of  
24 blocks needed exceeds the remainder of the quota.

25  
26 10. A method as in claim 1, further comprising the step of releasing reservation

1 of blocks as those blocks are written to storage.

2  
3 11. A method as in claim 10, wherein the step of releasing reservation of blocks  
4 further comprises the step of decrementing a reserved block count in a file system information  
5 block by a number of released blocks, the reserved block count indicating how many unallocated  
6 blocks have been reserved for files in the file server.

7  
8 12. A method of managing a file system for a file server, comprising the steps of:  
9 receiving a file operation that signals a reservation operation for a file for which  
10 reservation has already been performed, said reservation operation specifying a new file size  
11 different from a current file size for the file;  
12 comparing the current file size with the new file size;  
13 in the case that the current file size exceeds the new file size, releasing the  
14 the remaining block reservations for the file;  
15 in the case that the new file size exceeds the current file size, reserving in the file  
16 system an additional number of unallocated blocks equal to a difference between a total number  
17 of direct and indirect blocks required by the new file size and a total number of direct and  
18 indirect blocks required by the current file size.

19  
20 13. A method as in claim 12, wherein the file system uses a write anywhere file  
21 system layout.

22  
23 14. A method as in claim 12, wherein the file operation that signals the  
24 reservation operation is a zero length write request.

25  
26 15. A method as in claim 12, wherein the file operation that signals the

1 reservation operation includes a parameter that specifies the file size.

2  
3 16. A method as in claim 12, wherein the step of releasing remaining block  
4 reservations for the file further comprises the steps of:

5 resetting a flag in an inode for the file that indicates blocks have been reserved for  
6 the file; and

7 decrementing a reserved block count in a file system information block by a  
8 number of blocks still reserved for the file, the reserved block count indicating how many  
9 unallocated blocks have been reserved for files in the file system.

10  
11 17. A method as in claim 12, further comprising the step of checking that a  
12 number of available blocks in the file system is greater than the additional number of unallocated  
13 blocks, wherein an error is returned in a case that the number of available blocks is less than the  
14 additional number of blocks.

15  
16 18. A method as in claim 17, wherein the number of available blocks in the file  
17 system is determined by subtracting a number of allocated blocks, a number of cached  
18 unallocated blocks, and a number of reserved blocks from a total number of blocks in the file  
19 system, and adding a number of reserved cached unallocated blocks.

20  
21 19. A method according to claim 12, further comprising the step of checking that  
22 the additional number of blocks does not exceed a remainder of a quota for an owner of the file,  
23 wherein an error is returned in a case that the additional number of blocks exceeds the remainder  
24 of the quota.

25  
26 20. A method as in claim 12, further comprising the step of releasing reservation

1 of blocks as those blocks are written to storage.

2  
3 21. A method as in claim 20, wherein the step of releasing reservation of blocks  
4 further comprises the step of decrementing a reserved block count in a file system information  
5 block by a number of released blocks, the reserved block count indicating how many blocks total  
6 have been reserved for files in the file server.

7 //

8 //

9 //

10 //

11 //

12 //

13 //

14 //

15 //

16 //

17 //